# Passive Corruption in Statistical Multi-Party Computation

## (Extended Abstract)*

Martin Hirt[1], Christoph Lucas[1], Ueli Maurer[1], and Dominik Raub[2]

[1] Department of Computer Science, ETH Zurich, Switzerland
{hirt,clucas,maurer}@inf.ethz.ch
[2] Department of Computer Science, University of Århus, Denmark
raub@cs.au.dk

**Abstract.** The goal of *Multi-Party Computation* (MPC) is to perform an arbitrary computation in a distributed, private, and fault-tolerant way. For this purpose, a fixed set of $n$ parties runs a protocol that tolerates an adversary corrupting a subset of the parties, preserving certain security guarantees like correctness, secrecy, robustness, and fairness. Corruptions can be either *passive* or *active*: A passively corrupted party follows the protocol correctly, but the adversary learns the entire internal state of this party. An actively corrupted party is completely controlled by the adversary, and may deviate arbitrarily from the protocol. A *mixed adversary* may at the same time corrupt some parties actively and some additional parties passively.

In this work, we consider the statistical setting with mixed adversaries and study the exact consequences of active and passive corruptions on secrecy, correctness, robustness, and fairness separately (i.e., hybrid security). Clearly, the number of passive corruptions affects the thresholds for secrecy, while the number of active corruptions affects all thresholds. It turns out that in the statistical setting, the number of passive corruptions in particular also affects the threshold for correctness, i.e., in all protocols there are (tolerated) adversaries for which a single additional passive corruption is sufficient to break correctness. This is in contrast to both the perfect and the computational setting, where such an influence cannot be observed. Apparently, this effect arises from the use of information-theoretic signatures, which are part of most (if not all) statistical protocols.

**Keywords:** Multi-party computation, passive corruption, statistical security, hybrid security, mixed adversaries.

* The full version of this paper is available at the *Cryptology ePrint Archive*: http://eprint.iacr.org/2012/272. This work was partially supported by the Zurich Information Security Center.

A. Smith (Ed.): ICITS 2012, LNCS 7412, pp. 129–146, 2012.

# 1 Introduction

## 1.1 Secure Multi-Party Computation

*Multi-Party Computation* (MPC) allows a set of $n$ parties to securely perform an arbitrary computation in a distributed manner, where security means that secrecy of the inputs and correctness of the output are maintained even when some of the parties are dishonest. The dishonesty of parties is modeled with a central adversary who corrupts parties. The adversary can be *passive*, i.e. can read the internal state of the corrupted parties, or *active*, i.e., can make the corrupted parties deviate arbitrarily from the protocol.

MPC was originally proposed by Yao [Yao82]. The first general solution was provided in [GMW87], where, based on computational intractability assumptions, security against a passive adversary was achieved for $t < n$ corruptions, and security against an active adversary was achieved for $t < \frac{n}{2}$. Information-theoretic security was achieved in [BGW88, CCD88] at the price of lower corruption thresholds, namely $t < \frac{n}{2}$ for passive and $t < \frac{n}{3}$ for active adversaries. The latter bound can be improved to $t < \frac{n}{2}$ if both broadcast channels are assumed and a small error probability is tolerated [RB89, Bea89]. These results were generalized to the non-threshold setting, where the corruption capability of the adversary is not specified by a threshold $t$, but rather by a so called adversary structure $\mathcal{Z}$, a monotone collection of subsets of the player set, where the adversary can corrupt the players in one of these subsets [HM97].

All mentioned protocols achieve full security, i.e. secrecy, correctness, and robustness. *Secrecy* means that the adversary learns nothing about the honest parties' inputs and outputs (except, of course, for what can be derived from the corrupted parties' inputs and outputs). *Correctness* means that all parties either output the right value or no value at all. *Robustness* means that the adversary cannot prevent the honest parties from learning their respective outputs. This last requirement turns out to be very demanding. Therefore, relaxations of full security have been proposed, where robustness is replaced by weaker output guarantees: *Fairness* means that the adversary can possibly prevent the honest parties from learning their outputs, but then also the corrupted parties do not learn their outputs. *Agreement on abort* means that the adversary can possibly prevent honest parties from learning their output, even while corrupted parties learn their outputs, but then the honest parties at least reach agreement on this fact (and typically make no output). In our constructions, all abort decisions are based on publicly known values. Hence, we have agreement on abort for free.[1]

The traditional setting of MPC has been generalized in two directions. On the one hand, the notion of *hybrid security* was introduced to allow for protocols with different security guarantees depending on the number of corruptions [Cha89, FHHW03, FHW04, IKLP06, Kat07, LRM10, HLMR11]. Intuitively, the more corrupted parties, the less security is guaranteed. This model also allows to analyze each security guarantee separately and independent of other guarantees. On the other hand, protocols were presented that do not restrict the adversary to

[1] The impossibility proof holds even when agreement on abort is not required.